ISSN 1561-5359. Штучний інтелект, 2017, № 3-4

УДК 512.64

Є.О.Осадчий[1], О.Є.Осадчий[2], Р.В.Скуратовський[3]
[1]Київський національний університет імені Тараса Шевченка, Україна
вул. Володимирська, 60, м. Київ, 01033
[2] Університет Феникс, Аризона, США
4025 S. Riverpoint Parkway, Phoenix, AZ 85040
[3] Міжрегіональна академія управління персоналом, Україна
вул. Фрометівська, 2, м. Київ, 03039

ЧИСЛОВІ ЗАКОНОМІРНОСТІ ТА ТАЙМЕРНЕ КОДУВАННЯ ІНФОРМАЦІЇ

Y.O Osadchyy[1], O.Y. Osadchyy[2], R.V. Skuratovskii [3]
[1]Kyiv National University named by Taras Shevchenko, Ukraine
60, Volodymyrivska str., Kyiv, 01033
[2] University of Phoenix, Arizona, USA
4025 S. Riverpoint Parkway, Phoenix, AZ 85040
[3]Kyiv Interregional Academy of Personnel Management, Ukraine
2, Frometivska str., Kyiv, 03039

NUMERICAL REGULARITIES AND TIMER CODING OF INFORMATION

У статті пропонується стратегія розширення методів кодування інформації для уніфікації ідентифікації об’єктів пізнання. На основі відомих важливих числових закономірностей виявляються нові, що можуть знайти застосування в комп’ютерах майбутнього.

Ключові слова: кодування інформації, числові закономірності, ідентифікація об’єктів, таймерне кодування, цифровий комп’ютер.

The paper proposes a strategy of expanding information coding methods to unify identification of objects. Based on known important numerical regularities, new ones are found, which can be used in computers of the future.

Keywords: information coding, numerical patterns, object identification, timer coding, digital computer.

Вступ

Будь-які дані в цифровому комп’ютері є числом, що потенційно наділене властивістю кількості. Для двійкового комп’ютера найменшою одиницею даних є біт, а похідними – байт, запис, файл та інші. Базовою одиницею даних у ньому є байт. Його смислове перетворення в інформацію здійснюється через еталонну (ASCII) та похідні (КОІ-8, ISO 8859-5 та інші) таблиці кодування. Альтернативне «таймерне» кодування забезпечується використанням вбудованого в комп’ютер таймера. Він є незалежним двійковим лічильником (процесором), призначеним для супроводу, в часі, роботи основного процесора (процесорів). Його вплив забезпечується операцією «переривання по таймеру». У роботі пропонується стратегія розширення можливостей існуючих методів кодування для уніфікації ідентифікації об’єктів пізнання.

Постановка проблеми

Існує одна із важливих задач штучного інтелекту – оптимізація комп'ютерного перетворення (трансформації) інформації. Авторами запропоновано новий підхід до розв'язання цієї проблеми, який базується на двох видах кодування, що використовуються в комп’ютерах Фон-Нейманівської архітектури: двійкового та таймерного. Дослідники розглянули важливі числові закономірності архітектур

цифрових комп'ютерів, а також нові, що є перспективними в комп'ютерах майбутнього. Для обґрунтування використовуються методи та засоби реалізації реального комп'ютера одинкової (унарної) архітектури. Через числові закономірності це передбачив Тюрінг, а перспективність – дослідження інших [3-6].

Аналіз останніх досліджень і публікацій

Уперше використання унарної арифметики для обґрунтування принципів алгоритмізації запропонував Алан Метісон Тюрінг – англійський математик, логік, криптограф, що зробив істотний вплив на розвиток інформатики (зокрема, теорії штучного інтелекту). Запропоновану ним у 1936 році абстрактну обчислювальна машину[1], можна вважати моделлю комп'ютера загального призначення. Вона дозволила формалізувати поняття алгоритму і до цих пір використовується в безлічі теоретичних і практичних досліджень. Як криптограф, він також мав надзвичайні заслуги в перемозі над фашизмом. До сьогодні мало дослідженим є клас задач, дотичний до проблеми виявлення та використання числових закономірностей що є важливими для вирішення окреслених Тюрингом проблем інформатики та штучного інтелекту. Вперше можливість технічної реалізації основних складових комп'ютера унарної архітектури була доведена винаходами вітчизняних дослідників [3, 4]. Останній їхній винахід [5], на основі переваг виявлених числових закономірностей одиничного (унарного) кодування, здатен прискорити вирішення проблеми взаєморозуміння при використанні діалектів мови спілкування. Це дозволить, наприклад, прискорити створення універсальної мови спілкування глухих. Також вперше ними була запропонована математична модель ймовірнісно – статистичного, а не створюваного рахуванням, генератора текстів природної мови [6]. Виявлення кількісного значення числа з послідовності «1» унарної СЧ, порівняно з представленням в будь якій з похідних СЧ, потребує значно більшої кількості операцій. Тому, одиничне кодування рахуванням вимагає найбільший простір пам'яті. Разом з тим, положення останньої «1» в їх послідовності - потенційно визначає значення кількості числа. Немає необхідності підраховувати всі «1», коли значення числа зберігається в стабільно визначеній позиції пам'яті. Достатньо отримати лише цю ознаку, щоб, наприклад, аналітично, визначити значення основної ознаки числа – його кількість. Це дозволяє стверджувати, що ця СЧ в дійсності є змішаною «одинично - n річною СЧ». В подальшому, для скорочення, її позначимо як «одинкову» СЧ. Нижче, дослідниками доводиться, що на основі одинкової арифметики можливим є створення цифрового комп'ютера з унікальними технічними параметрами. Обмежимось показаними результатами так як останні суттєві досягнення інших розробників в цій галузі нам невідомі.

Виклад основного матеріалу

З метою однозначного розуміння подальшого змісту приймемо, що цифрова інформація, за замовчуванням, відображається в десятирічній системі числення (СЧ). При необхідності показати іншу СЧ її основа писатиметься індексом. Еталонне кодування ASCII займає першу половину байту двійкової таблиці кодування і містить кількісне значення послідовності з перестановок 4 розрядних чисел в $СЧ_{(2)}$. Кожному з чисел двійкової послідовності поставлено в відповідність унікальний символ алфавіту мов спілкування з комп'ютером, кожному з яких відповідає своя СЧ. Як правило, використовується обмежена багаторівнева комбінаторика символів алфавіту мови спілкування. Адекватна обмежена лінійна комбінаторика двійкових кодів ASCII дає можливість керувати діями комп'ютера (програмою) та спілкуватись, в т. ч.

природною мовою зі співбесідниками (користувачами). Назвемо таку усвідомлену для користувачів послідовність байтів – повідомленням. У бітах пам’яті комп’ютера воно виглядає натуральним двійковим числом великої розмірності. Для зручності обробки інформації його структурують у вигляді (байта, запису, файлу і т.д.). Закономірність побудови заданої послідовності чисел визначають СЧ або мови, що для цього застосовуються. Для визначення використовуваних нами понять скористаємось Бекус-Наура формальною системою. Обмежимось аналізом закладеної в ASCII можливості смислової текстової інтерпретації.

<таймерна мітка>::=<визначний символ >|<час генерації числа>

<час генерації числа >::='відтворена лічильником кількість цифрами'

<визначний символ >::= 'незадіяний у повідомленні символ алфавіту'

<система числення>::=<мова >|<основа>|<СЧ$_{(i)}$ >

<основа>::=<кількість символів>|< СЧ$_{(i)}$ >|<цифра>

<символ >::=<буква>|<число без знаку>{<символ>|<число без знаку>}

<число без знаку>::=<цифра>{< цифра >}

<число >::=<цифра >{<цифра>}

<мова >::=<алфавіт>|< повідомлення >

<повідомлення>::=< символ алфавіту>{< символ алфавіту>< символ алфавіту>}

<алфавіт>::=<символ>{<символ>}

<символ >::=< байт >

<байт>::=<знак>|буква>|<цифра>

<знак>::=<розділовий>|<керуючий>|<числа>

<розділовий знак>::='_'|','|'.'|';'|':'|'-'|';'|'?'

<керуючий знак>::=<слово>|<знак>

<буква>::='A'|'B'|'C'|'D'|'E'|'F'|'G'|'H'…'Z'

<слово>::=<буква>|<цифра>|{<буква><цифра>}|{<цифра >< буква>};
{< буква>|<цифра>|<буква><цифра>|<цифра >< буква>};'SOH'|'STX'|…|'STX'

<знак числа>::='-'|'+'|'='

<цифра >::='1'|'2'|'3'…n

Таким чином, нами встановлено, що при наданні числу (даним) смислових ознак воно перетворюється в інформацію. Ще було встановлено, що значення числа може бути представлено як кодом (потенційною), так і актуалізованою послідовністю похідних цифр, наприклад, отриманих рахуванням. Ми означили лише ті поняття, що не були показані в посиланнях. Ефективним використанням виявленої закономірності числа є наявність у комп’ютері оперативної (прямого доступу) та лінійної пам’яті. Така пам’ять може бути як довготривалою (стійкою), так і одноразового застосування. Та найбільш ємною та основною, в цифрових комп’ютерах, є лінійна пам’ять. Кожна з видів пам’яті також має свої недоліки та переваги. Покажемо, що комп’ютер, що базується на будь-якій позиційній СЧ, крім одиничної (унарної), має значні часові втрати на обробку інформації великого обсягу та технічні складності в реалізації архітектури. Наш підхід до створення пам’яті прямого доступу на базі одиничної арифметики відображений у винаходах, наприклад, в [4]. Його перевага перед аналогами обґрунтовується тим, що відображенню кількісного значення числа, порівняно з його представленням у будь-якій з похідних СЧ, відповідає позиція останнього символу. Інші СЧ, у зв’язку з використанням у них неповної комбінаторики символів алфавіту, потребують складних структурних та алгоритмічних перетворень СЧ, що використовуються.

Тому, хоча одиничне кодування вимагає більшого простору лінійної пам’яті, прямий доступ до останньої його позиції дозволяє найпростіше визначати та використовувати кількісне значення повідомлення. Двійкове кодування в ASCII вимагає використання повної комбінаторики обмеженої кількістю розрядів. Кожному з двійкових значень відповідає своя комбінаторика символів багатьох похідних СЧ (мов спілкування). Комунікатори спілкуються через цю таблицю. Для спілкування з людиною створюються периферійні пристрої. Сьогодні вони орієнтовані на використання майже всіх відомих органів чуття: зір, слух, смак, нюх(запах), тактильного сприйняття (дотик, відчуття болю, температури), просторової орієнтації (відчуття: рівноваги, ваги; положення в просторі; прискорення). Найбільш розповсюдженими є клавіатура та монітор. Кожен з периферійних пристроїв має свої переваги та недоліки, та вони не є об’єктами нашого дослідження. Закладені через похідні СЧ, в кодувальних таблицях, цифрові закономірності повинні враховувати існуючі статистично – ймовірнісні пріоритети предметної галузі. Вони є дуже важливими для оптимізації комп’ютерного перетворення (трансформації) інформації предметної галузі, що осмислюється. Розглянемо інші, крім кількісних, корисні властивості деяких послідовних закономірностей натуральних чисел та виявимо ознаки, що їх ідентифікують.

Числова послідовність Фібоначчі: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89... Кожне наступне число є $\sum$ двох попередніх, саме така послідовність називається послідовністю Фібоначчі, і визначається вона такими рекурентними співвідношеннями:

$$u_1 = 1, u_2 = 1,\ u_{n+1} = u_n + u_{n-1},\ n \geq 2\,.$$

Така послідовність цілих чисел названа «Золотим перерізом», тому що виявлені закономірності пропорцій частин спірального напрямку еволюційного розвитку множин стійких об’єктів природи в пропорції (приблизно 1,615). Провівши отримані пропорції вимірів та співвідношень, наприклад, між частинами тіла людини, можна зробити висновок, що середнє значення пропорції частин тіла коливається в межах співвідношення 42 : 26 = 1,615. На жаль, виявлена закономірність не є однозначною для цифрового кодування універсального двійкового комп’ютера.

Закономірність отримання значення кількості для двійкового числа в комп’ютері.

Незалежно від СЧ, у якій записано смислове значення числа, проаналізуємо існуючу закономірність відображення натурального числа в комп’ютері. Відомо, що перетворення простого числа в двійковому комп’ютері розпочинається з операції зсуву (переміщення). Далі йдуть операції додавання, віднімання, множення, ділення й інші математичні та логічні перетворення числа. Для її аналізу скористаємось одиничним(унарним) кодуванням, бо воно найпростіше показує кількісний зміст двійкових бітів будь-якого байту комп’ютера. Ми вибрали таку СЧ ще й тому, що за визначенням Тюрінга [1] саме вона є найпростішою. Це має значення і тому, що тільки для СЧ$_{(1)}$ не є суттєвим, яка з можливих цифрових закономірностей нами буде використовуватись для отримання кількісного значення числа. Адже тільки така СЧ дозволяє дуже просто перейти на будь-яку з можливих, у тому числі і ту, що створюється довільним генератором повідомлення, а не тільки визначеними наперед закономірностями. Розпочнемо аналіз з найбільш важливих закономірностей двійкової СЧ. Відомо, що ключовими для переходу в старший розряд двійкової СЧ$_{(2)}$ є наявність наступного значення «1» в послідовностях бітів у байті:$1_{(2)}$, $11_{(2)}$, $111_{(2)}$, $1111_{(2)}$, …111...$1_{(2)}$. Це говорить про те, що кожен наступний розряд двійкового числа

є числом $2_{(10)}$, у степені, що має бути записана у визначеній байтом кількості задіяних біт. Так як просте число «0» не містить значення кількості, почнемо рахувати з 1. Не використовуючи інших символів (цифр) продовжимо рахувати числа в «1»(унарній) СЧ. У результаті застосування $СЧ_{(1)}$, отримаємо таку кількість 1, що відповідає кількісному визначенню значення структурованого байтом числа. Тоді, в комфортній для людини десятковій СЧ матимемо наступні кількісні значення цієї послідовності з 1 бітів у байті: $1_{(10)}$, $3_{(10)}$, $7_{(10)}$, …, $n\text{-}1_{(10)}$, $128_{(10)}$.). Цю властивість ми назвемо ***обмеженістю зміни комбінації*** символів. Використовуючи її, наприклад, зможемо за визначеними кількісними ознаками позицій (розрядів) ідентифікувати кількісне значення будь-якого двійкового числа байта. Для цього скористаємось операціями «+», «-» та операцією порожнього символу (позиції, значення якої не розпізнається). Останнє означає, що виявлене значення кількості доповнюється значеннями всіх попередніх розрядів, що перетворює його в їх послідовність. В одиничній арифметиці цим діям відповідає переміщення на один розряд вправо або вліво. Відсутність переміщення в $СЧ_{(1)}$ означає порожню операцію, яка рівнозначна тому, що виявлена позиція 1 доповнюється значеннями всіх попередніх 1 символів, тобто перетворюється в їх послідовність.

Виявлена закономірність є справедливою і для інших СЧ. Наприклад, у трійковій СЧ, що також структурована байтами, отримаємо: $2_{(3)}$, $22_{(3)}$, $222_{(3)}$, $2222_{(3)}$,…,$222…2_{(3)}$. Тут, у першій комірці (чарунці, біті) буде виявлено кількість $3_{(10)}$. У послідовності з двох $–7_{(10)}$ і т.д. Нарешті, в «n» річній СЧ, яка відповідає системі числення з 1 символу, що не допускає ніяких комбінацій. У такій СЧ може бути записане тільки єдине кількісне значення числа і воно може бути будь-яким, але наперед визначеним. У випадку ASCII це буде число $256_{(10)}$ з розумінням кількості, що його відтворює. На прикладі 1 і n річної СЧ. Можна помітити, що одна і та ж кількість може бути визначена як генерацією, наприклад, рахунком (у визначеній СЧ), так і його безпосереднім значенням, що реалізоване положенням останньої позиції числа. Виявлені нами закономірності є близькими до закономірності «золотого перетину», але майбутнім дослідникам ще є до чого рухатись. З нашої точки зору, ідеальною закономірністю буде така, що забезпечить миттєву переробку необмеженої кількості інформації. Теоретично, це дозволяє лише трансформерна технологія таймерного кодування інформації для комп’ютера одинкової архітектури, що вперше було доведено в [3].

У сьогоднішніх умовах масового використання двійкового комп’ютера слід використовувати реальні можливості його архітектури. У ній вже закладені ключові властивості згаданої технології – миттєвий доступ та зберігання великих обсягів інформації. Технічно, вони реалізуються використанням у комп’ютері двох видів пам’яті: лінійної та прямого доступу. Таймерне кодування інформації, як альтернативне, частково реалізоване використанням додаткового паралельного процесора – таймера та операції «переривання по таймеру». Розглянемо похідні, від наведених, корисні можливості. Перевага лінійної пам’яті в тому, що пошкоджене число записане в ній завжди може бути відновлене хоча б приблизно або повністю. Наприклад, стискуючи в файлі, записаному в $СЧ_{(1)}$, значення всіх попередніх, крім останнього, байтів, тому що вони є найбільш придатними для цього. Недоліком такої пам’яті є значна втрата часу на відтворення значення інформації в розумінні прийнятою мовою спілкування. Особливо затратною в часі є відновлення кількості числа в $СЧ_{(n-1)}$. Навіть у $СЧ_{(2)}$ кожен наступний розряд вимагатиме 2-степеневого

збільшення часу генерації (наприклад, при рахуванні значення числа). Тому, при лінійному записі числа (у вигляді послідовності символів СЧ) дійсною є наступна закономірність. Чим більше символів у повідомленні, тим довшим є процес визначення його кількості (значення числа). Перевага розміщенню числа в пам'яті прямого доступу. У ній кількісне значення числа визначається миттєво (за 1 ітерацію), та тут натуральне число не може бути великим і є найбільш незахищеним від пошкодження ще й тому, що така пам'ять переважно є одноразовою або нестійкою від пошкодження. Задіяні додаткові методи перетворення двійкового числа стисненням або паралельними обчислюваннями також мають непереборні обмеження. Тому, коли пошкоджена архівована інформація, її відновити без втрат майже неможливо. Існуючі методи збереження в лінійній пам'яті значення числа значно збільшують його розмір, а технічна реалізація пам'яті прямого доступу є, порівняно з послідовною, занадто затратною. Зрозуміло чому, в будь-якому комп'ютері її розмір є обмеженим. Відповідно до архітектури двійкового комп'ютера прямий доступ до таблиць кодування, в першу чергу, ASCII надав йому «живучості», в плані можливості універсального застосування в реалізації потрібних алгоритмів. Це пояснюється тим, що в цій таблиці міститься вся можлива комбінаторика 4-розрядного двійкового числа еталонної частини таблиці і адекватного значення: його кодам необхідне смислове навантаження їх можливого використання. Тому, за умови обов'язкової присутності цієї частини таблиці в будь-якому комп'ютері, можна вважати, що її зміст можна актуалізувати посиланням єдиного таймерного символу. Далі цей процес можна літерувати наступним чином: при досягнені зменшення кількості значущих байт до встановленої нами межі переходимо до наступної ітерації стиснення. Замінюємо значення таймерної мітки змістом одного біта у вигляді «1». У результаті зменшується просторовий розмір файлу. Коли всі заміни будуть проведені отримаємо файл, що міститиме залишкові «значущі» байти та двійкове число. Останнє знову підлягає таймерному кодуванню. Виконаємо перехід від одиничної системи числення, через ознаку зупинки генератора повідомлення, закодовану одним бітом у вигляді «1». Для цього, перетворимо її у відповідне число, що є основою n-ричної системи числення. Базисну послідовність відповідної СЧ будемо обирати так, щоб запис саме цього числа був найкоротшим. Для цього візьмемо $n = \#\{1\}_{\mu(T)} = lenght\,\mu(T)$. Саме одинична $СЧ_{(1)}$ ефективно використовується у таймерній машині Тюрінга, де її також можна назвати унарною, бо для запису команд потрібно лише значення «зміст» таймерної мітки, яке представляється одним бітом у вигляді «1». Форма запису правила таймерної машини Тюрінга «$q_i a_j \rightarrow q_{i1} a_{j1} R/L/N$» за допомогою $СЧ_{(1)}$ реалізується як перебування у певному стані $q_i,\ i \in \{0, 1\}$ таймерного генератора. Можливі наступні його стани: запис таймерної мітки t_1, або зчитування t_2 і також – очікування t_0. Результат функції запису таймерної мітки a_{j1} в комірці на стрічці відповідної машини Тюрінга позначений у команді як a_{j1}. Далі в формі маємо зсув на 1 позицію вправо чи вліво, що кодується відповідно R/L, ситуація відсутності зсуву позначається як N. Вибір команди зсуву записаний у комірці разом з інформацією і позначається як таймерна мітка a_j. Двійкова СЧ, яка використовується у машині Тюрінга, одразу не дає результату в одну комірку безмежної стрічки. Другим недоліком класичної машини Тюрінга є неможливість використання операції зсуву з максимальною ефективністю. Це тому, що через обмеженість розміру *k* комірки

маємо не більше ніж 2^k комбінацій. Звідси перехід до наступної комірки дає не більше k за інформативних бітів, на відміну від таймерної машини Тюрінга, де така зміна не обмежена. Будемо вважати, що наведених доказів є достатньо для показу переваги унарної СЧ перед іншими та можливість її застосування в реалізації таймерної машини Тюрінга навіть без використання операцій двійкової арифметики.

Відомо, що базовою операцією комп'ютера фон-Нейманівської архітектури є двійкове додавання. Відповідно, базовим конструктивним елементом – тригер. Основним недоліком таких комп'ютерів є обмеження швидкодії при обробці великих обсягів інформації. Тому, об'єктивно існуюча теоретична межа стиснення двійкової інформації вже практично досягнута в сучасних програмах-архіваторах. Аналогічно, є непереборним обмеження в технічній реалізації двійкової архітектури. Особливо це відчутно в напрямках прискорення передачі інформації великого розміру по лінійним каналам комунікації, та й швидкодія двійкових процесорів, навіть в умовах надпровідності, обмежена. Методи та засоби розпаралелювання і розподіленої обробки інформації не є панацеєю. Існуючий напрямок створення квантового комп'ютера має перспективу, тому що використовує означений нами стан невизначеності значення біта, але, поки що, для існуючого рівня розвитку технічного прогресу він не є реальністю.

Висновки

Реальною альтернативою, що дає можливість позбутись існуючих обмежень двійкового комп'ютера, може стати запропонований напрямок їх покращення, що має завершитись створенням комп'ютерів одинкової архітектури. Їх перевага в теоретичній можливості обробки даних будь-якого розміру та реальній перспективі досягнення об'єктивно існуючої межі швидкодії його процесора. Вона сьогодні обмежена лише швидкістю руху електрона. Задекларование є можливим тому, що базовою операцією такого комп'ютера є операція зсуву (переміщення). Відповідно базовим технічним елементом для неї – лінія затримки, найпростішим варіантом якої є електричний опір. Базова одинкова математика є також надзвичайно простою і частково була визнана Аланом Тюрінгом [1] та іншими засновниками комп'ютеризації. Більш детально, разом з основними конструктивними елементами, вона розкривається в інших наведених нами посиланнях. За результатами даного дослідження готується новий патент на винахід, що ще раз підтвердить динаміку розвитку технічної реалізації та перспективність емуляції наведених методів та засобів одинкового комп'ютера, як одного з найперспективніших в обробці інформації.

Література

1. Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction // Proceedings of the London Mathematical Society – 1938. – Vol. s2-43, Iss. 6. – P. 544–546. – ISSN 0024-6115; 1460-244X – doi:10.1112/PLMS/S2-43.6.544
2. Зайцев Д.А. Математичні моделі дискретних систем: Навч.посібник. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – 40 с.
3. Осадчий Є.О. Трансформерні технології побудови машин і механізмів. – К.: Науковий світ, 2004.- 167 с.
4. А. с. 1747944 СССР, №1462420 МКИ5 G 11C 15/00. Ассоциативное запоминающее устройство / Осадчий Е.А., Зеебауер М., Марковский А.П., Корнейчук В.И., Галилейский Ф.Ф.- Опубл.1.11. 1988. – Бюл. №8. – 11 с.
5. Патент України на корисну модель № u 121740, МПК G 06 F 15/38 Пристрій для перетворення кодів з однієї мови на іншу / Крак Ю.В., Терещенко В.М., Осадчий Є.О., Горбунов О.А.- Опубл. 11.12.17. - Бюл. № 23.- 7 с.
6. Метод быстрого таймерного кодирования текстов / Р.В. Скуратовский // Кибернетика и системный анализ. – 2013. – Т. 49, № 1. – С. 154-160.

Literatura

1. Turing A.M. On Computable Numbers, with an Application to the Entscheidungsproblem. A Correction // Proceedings of the London Mathematical Society – 1938. – Vol. s2-43, Iss. 6. – P. 544–546. – ISSN 0024-6115; 1460-244X – doi:10.1112/PLMS/S2-43.6.544
2. Zaitsev D.A. Matematychni modeli dyskretnykh system: Navch.posibnyk. – Odesa: ONAZ im. O.S. Popova, 2004. – 40 s.
3. Osadchyi Ye.O. Transformerni tekhnolohii pobudovy mashyn i mekhanizmiv.- K.: Naukovyi svit, 2004. – 167 s.
4. A. s. 1747944 SSSR. №1462420 MKI5 G 11S 15/00. Assotsiativnoye zapominayushcheye ustroystvo / Osadchiy E.A., Zeyebauyer M., Markovskiy A.P., Korneychuk V.I., Galileyskiy F.F.- Opubl.1.11. 1988.- Byul. №8.- 11 s.
5. Patent Ukrainy na korysnu model № u 121740, MPK G 06 F 15/38 Prystrii dlia peretvorennia kodiv z odniiei movy na inshu / Krak Yu.V., Tereshchenko V.M., Osadchyi Ye.O., Horbunov O.A. – Opubl. 11.12.17. – Biul. № 23. – 7 s.
6. Metod bystrogo taymernogo kodirovaniya tekstov / R.V. Skuratovskiy // Kibernetika i sistemnyy analiz. – 2013. – T. 49. № 1. – S. 154-160.

RESUME

Y.A. Osadchyy, A.Y. Osadchyy, R.V. Skuratovskii

Numerical regularities and timer coding of information

The paper proper is strategy is strategy for expanding the coding of information for identification of objects. Because of known important numerical regularities, new ones can be found which can be used in computers of the future. A real alternative, which allows you to get rid of the existing limitations of a binary computer, can be the proposed direction of their improvement, which should complete the creation of computers for a single architecture. Their advantage in the theoretical ability to process data of any size and real prospect achieve an objectively existing limit of the speed of its processor. It is today limited only by the velocity of the electron. Declared possible because the main operation of such a computer is a shift operation (shift). The corresponding basic technical element for it is a delay line, the simplest of which is electric resistance. Basic singular mathematics for them was also extremely simple and partly recognized by Alan Thuringia and other founders of computerization. Creation of single and double counters emulations programs. The comparison of programs shows the real advantages and disadvantages of linear (position) and timer (mixed) encoding. According to the results of this study, new patents for inventions are being prepared. The latest invention of authors, based on the advantages of the detected numerical laws of unit (universal) coding, is capable of accelerating the solution of the problem of mutual understanding when using the dialects of the language of communication. This will, for example, accelerate the creation of a universal language for communication of the deaf. Also, for the first time a mathematical model was proposed - statistically, rather than created, a natural language text generator. This confirms the growing dynamics of the technical and program realization of the mentioned methods and means of a single computer, as one of the most promising in processing information.

Надійшла до редакції 20.10.2017

